**(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)**

AB

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
8 March 2001 (08.03.2001)

PCT

(10) International Publication Number
WO 01/17152 A1

(51) International Patent Classification[7]: **H04K 1/04**

(21) International Application Number: PCT/BR99/00076

(22) International Filing Date:
13 September 1999 (13.09.1999)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PI 9903609-6   27 August 1999 (27.08.1999)   BR

(71) Applicant *(for all designated States except US)*: **COPPE/UFRJ - COORDENAÇÃO DOS PROGRAMAS DE PÓS GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO** [BR/BR]; Centro de Tecnologia, S/Nº, Bloco G, Ilha do Fundão, CEP-20000-000 Rio de Janeiro, RJ (BR).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **CARDOSO SALOMÃO, Sérgio, Luiz** [BR/BR]; Rua Torres Sobrinho, 56, B1II apto, 204, Méier, CEP-20780-050 Rio de Janeiro,

RJ (BR). **CASTRO ALVES, Vladimir** [BR/BR]; Rua o9, Quadra 87, Lote 5A, casa 12, Camboinhas, Niterói, CEP-24358-000 Rio de Janeiro, RJ (BR). **MONTEIRO CHAVES, Filho, Eliseu** [BR/BR]; Av. Mal. Henrique Lott, 270, apto 1104, Barra da Tijuca, CEP-22631-370 Rio de Janeiro, RJ (BR).

(74) Agents: **JOUBERT GONCALVES DE CASTRO & ZULDECH ASSESSORIA EMPRESARIAL LTDA**; Av. Ernani do Amaral Peixoto, 458 s1001, Centro Niterói, CEP-24020-077 Rio de Janeiro, RJ et al. (BR).

(81) Designated States *(national)*: AL, AM, AT, AU, AZ, BA, BB, BG, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: A METHOD FOR THE HARDWARE IMPLEMENTATION OF THE IDEA CRYPTOGRAPHIC ALGORITHM - HIPCRYPTO
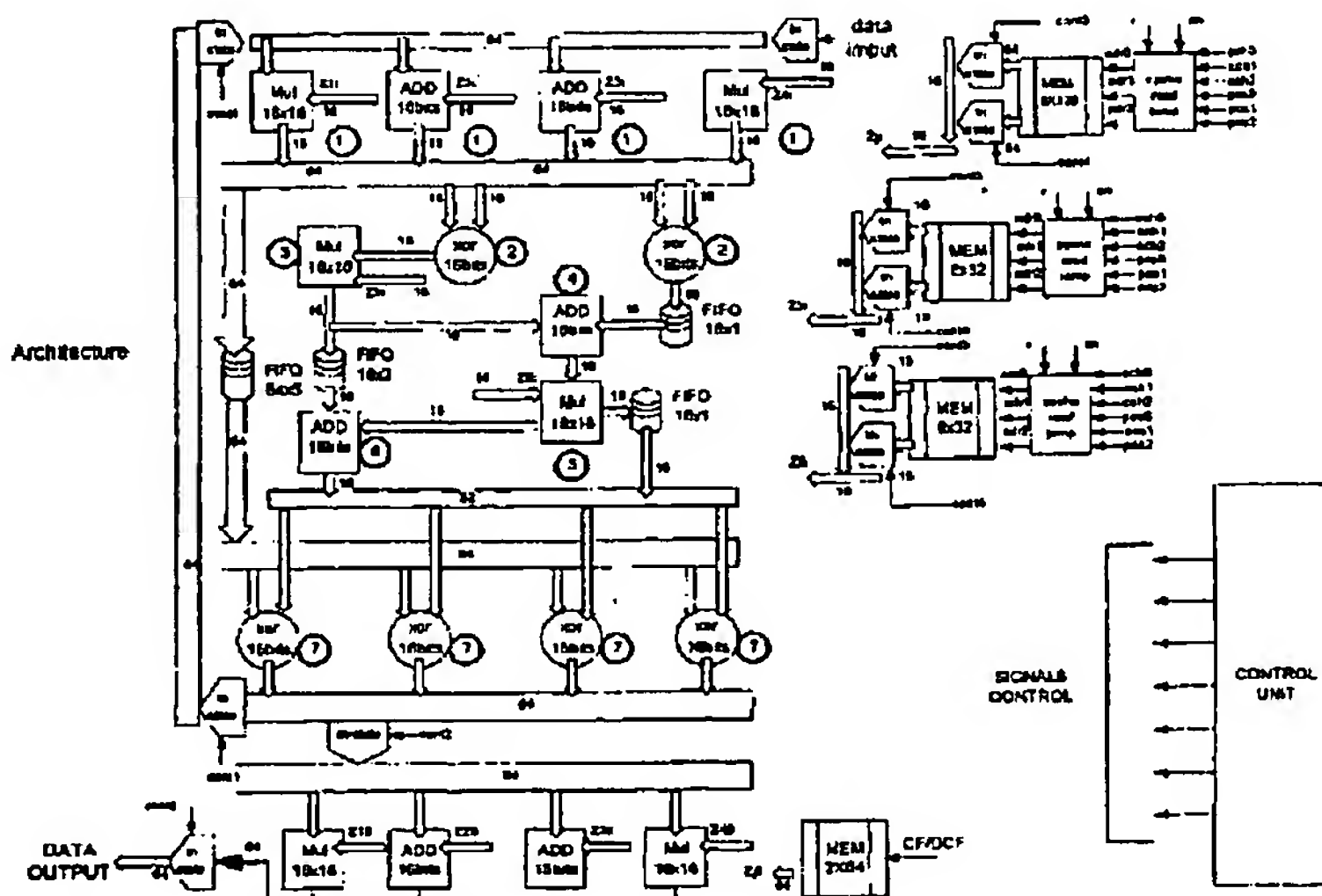
(57) Abstract: HiPCrypto is the hardware implementation of the most secure private key cryptographic algorithm: IDEA (International Data Encryption Algorithm), through the exploitation of spatial and temporal parallelism techniques in order to achieve the processing speed required by ATM and Gigabit networks.

WO 01/17152 A1

WO 01/17152 A1

Published:
— *With international search report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Description of the Invention "A METHOD FOR THE HARDWARE IMPLEMENTATION OF THE IDEA CRYPTOGRAPHIC ALGORITHM - HiPCrypto"

TECHNICAL FIELD

5   HIPCrypto is a hardware architecture proposal for the IDEA cryptographic algorithm, in which were used techniques for the exploitation of spatial and temporal parallelism, in order to reach the processing speeds required by real time applications and high speed data
10 communication networks such as ATM.

Nowadays, a world tendency exists for the use of networks that provide different types of Telecommunication services such as the Integrated Service Data Network (ISDN). These types of networks should provide
15 a wide range of services from telephone and cable TV to video conference.

The technological progress of transmission data networks pushed the development of cryptographic algorithm that became progressively more complex and
20 robust. They are widely used by private and governmental organizations as well as individuals that need to ensure secrecy in data communication.

The increasing complexity of recent cryptographic algorithms require high processing
25 capabilities due to the large number of arithmetic and logic operations that have to be executed, in some cases for real time applications like in video confereces.

PREVIOUS TECHNIQUES

Direct hardware implementation of
30 cryptographic algorithms can ensure high processing speeds required by current and future applications in data

transmission and eliminate a potential bottleneck in data communication networks that require high security levels.

Consequently, several cryptographic algorithms were totally or partially implemented as
5 Application Specific Integrated Circuits.

Several hardware and software implementations have been developed in the past decade for the Data Encryption Standard (DES), the most popular private key cryptographic algorithm. Table 1 shows the performance
10 obtained for some software implementations in different platforms. Table 2 shows the performance obtained for some dedicated hardware implementations. From table 2 one can see that the 6868 integrated circuit from VLSI Technology reaches up to 512Mbit/s, which is not sufficient to support
15 some high end ATM applications. Futhermore, it cryptanalysis on DES proved that it is weaker than some recent private key cryptographic algorithms like IDEA. Few hardware implementations of IDEA or its predecessors were reported in the litterature. For example, an ASIC that
20 implements the PES algorithm, which originated IDEA, has reached up to 55 Mbits/s at 25 MHz.

DETAILED DESCRIPTION

IDEA cryptographic algorithm

The first form of the IDEA algorithm, was
25 created by " Xuejia Lai and James Massey " in 1990 (US05214703 patent) and was called PES (Proposed Encryption Standard). In 1991, the algorithm was strengthened and was called IPES (Improved Proposed Standard Encryption). In 1992 IPES was called IDEA (International Data Encryption
30 Algorithm), and is actually considered by many specialists

in the field of cryptography as the strongest existing symmetrical algorithm.

IDEA is a symmetric, block-oriented cryptographic algorithm, which uses 128-bit keys (thus making it practically immune to brute-force attacks) and 64-bit plaintext blocks. IDEA is build upon a basic function, which is iterated multiple times. As shown in Figure 1 the basic function is iterated eight times. The first iteration operates on the input 64-bit plaintext block and the successive iterations operate on the 64-bit block obtained from the previous iteration. After the last iteration, a final transformation step produces the 64-bit ciphertext block.

Figure 1 shows the structure of the basic function. It involves three simple operations: bitwise exclusive-or, addition modulo 2^{16} (addition, ignoring the " overflow ") and multiplication modulo $2^{16} + 1$ (multiplication, ignoring the " overflow "). For each iteration, the 64-bit input block is divided into four 16-bitsub-blocks. In Figure 1, X1, X2, X3 and X4 denote the four 16-bit input sub-blocks used by the each iteration. The 64-bit block produced by each iteration is also constituted by four 16-bit sub-blocks. In Figure 1, Y1(i), Y2(i), Y3(i) and Y4(i) denote the four sub-blocks resulting from the each iteration. The 128-bit key is divided into 52 16-bit sub-keys (sub-key generation is discussed ahead). Six sub-keys are used in each iteration and four sub-keys are used in the final transformation. In Figure 1, Z1(i), Z2(i), Z3(i), Z4(i), Z5(i) and Z6(i) denote the six sub-keys used in each iteration. The operations performed in the each iteration are:

1. Multiply sub-block $X1(i)$ by sub-key $Z1(i)$

2. Add sub-block $X2(i)$ and sub-key $Z2(i)$

3. Add sub-block $X3(i)$ and sub-key $Z3(i)$

4. Multiply sub-block $X4(i)$ by sub-key $Z4(i)$

5. XOR the results of (1) and (3)

6. XOR the results of (2) and (4)

7. Multiply the result of (5) by sub-key $Z5(i)$

8. Add the results of (6) and (7)

9. Multiply the result of (8) by sub-key $Z6(i)$

10. Add the results of (7) and (9)

11. XOR the results of (1) and (9)

12. XOR the results of (3) and (9)

13. XOR the results of (2) and (10)

14. XOR the results of (4) and (10)

The outputs of the iteration are the four sub-blocks produced by steps (11) to (14). The two inner sub-blocks from steps (12) and (13), $Y2(i)$ to $Y3(i)$, are swapped, except for the last iteration.

Figure 1 shows the structure of the final transformation. In this figure, $Z1(9)$ to $Z4(9)$ denote the four 16-bit sub-keys and Y1 to Y4 denote the four 16-bit sub-blocks of the 64-bit ciphertext block. The operations performed in the final transformation are:

15. Multiply sub-block X1 by sub-key $Z1(9)$ to obtain Y1

16. Add sub-block X2 and sub-key $Z2(9)$ to obtain Y2

17. Add sub-block X3 and sub-key $Z3(9)$ to obtain Y3

18. Multiply sub-block X4 by sub-key $Z4(9)$ to obtain Y4

The encryption and decryption sub-keys are generated from the single 128-bit key. Encryption sub-keys are generated as follows. Initially, the 128-bit key is

divided into eight 16-bit sub-keys. Six of these sub-keys, $Z1(1)$ to $Z6(1)$, are used in the first iteration. The two remaining sub-keys, $Z1(2)$ and $Z2(2)$, are for the second iteration. The original 128-bit key is then rotated left by

5   25 bits and the resulting key is again divided into eight 16-bit sub-keys. Four sub-keys, $Z3(2)$ to $Z6(2)$, are grouped with $Z1(2)$ and $Z2(2)$ and destined to the second iteration. The other four sub-keys, $Z1(3)$ to $Z4(3)$, are to be used in the third iteration. Next, the key is again rotated left by

10   25 bits, divided into eight 16-bit sub-keys and these sub-keys are grouped properly. This process is repeated each of the sub-keys for the eight iterations and for the final transformation have been generated. Decryption sub-keys are calculated as either the additive or the multiplicative

15   inverses of the encryption keys.

As stated, the main goal in designing HiPCrypto is to obtain a device which would meet the performance requirements of applications in current and future high-speed data networks. This was achieved by

20   including parallel execution techniques into the design of HiPCrypto's architecture. There are two opportunities for exploiting parallelism in the IDEA algorithm: in the execution of its basic function and in the iterations of this function.

25   Examining the data flow shown in Figure 1, one can identify groups of operations that are data independent. In each group, one operation does not use the results produced by other operations in the group. The sets of independent operations are: the multiply and add

30   operations in steps (1) to (4); the exclusive-or operations in steps (5) and (6); and the exclusive-or operations in

steps (11) to (14). These independent operations can be performed simultaneously, provided the architecture incorporates multiple functional units dedicated to the execution of each of them.

5   By including multiple functional units in the architecture, we are making use of spatial parallelism. Temporal parallelism can also be employed in the execution of the basic function, by overlapping in time the operations upon distinct plaintext blocks. In this way, 10   multiple blocks can be encrypted (or decrypted) simultaneously, instead of sequentially. This temporal parallelism was implemented with the pipeline shown in Figure 2.

           Stage 1 contains two add and two multiply 15   units that perform in parallel the independent operations in steps (1) to (4) of the algorithm.

           Stage 2 contains two exclusive-or units to execute the operations in steps (5) and (6) in parallel.

           Stages 3, 4, 5 and 6 contain a single add or 20   multiply unit and they execute, respectively, the operations in steps (7), (8), (9) and (10) of the algorithm.

           Stage 7 has four exclusive-or units to execute steps (11) to (14) in parallel.

25   The last stage has two add units and two multiply units and performs the algorithm's final transformation (see Figure 2). This stage will be referred to as the output stage.

           In Figure 2, one can notice the inclusion of 30   between stages of the pipeline. These queues temporarily hold data forwarded between non-adjacent stages. For

instance, stage 7 operates on sub-blocks from stages 1 and
5 (see Figure 1).

    A sub-block from stage 1 arrives five cycles
before the corresponding sub-block from stage 5, and
5   during this time interval it remains in one of the queues
connecting stages 1 and 7. When the sub-block from stage 5
is available, the sub-block in the front of the queue is
dequeued and paired with the sub-block from stage 5. A
queue is needed along the shortest path (in number of
10 stages) between two non-neighbor stages. The size of each
queue is indicated in figure 2.

    The final aspect in HiPCrypto's architecture
concerns the .generation and storage of the sub-keys. To
generate the encryption sub-keys, it would be necessary a
15 circuitry for the rotation and sub-division of the 128-key.
Moreover, the generation of the decryption sub-keys would
require an arithmetic unit for the calculation of additive
and multiplicative inverses. The inclusion of this
additional hardware would only be reasonable if the key
20 changes very frequently, say, every few blocks. But that is
not the common case in a private-key cryptosystem:
typically, the key shared by a group of partners is changed
in a long term basis (days or weeks, for example). For this
reason, only sub-key storage is provided. Sub-keys are
25 generated externally by the host system, and then
downloaded into the chip.

Architecture of HIPCrypto

    The HIPCrypto architecture, Figure 3,
executes a complete iteration of the algorithm. This
30 architecture is composed of six 16-bit multipliers, six

16-bit adders and six 16-bit exclusive-or, memories for sub-key storage, buffers, tri-states and a control unit.

The operations contained in each stage of the pipeline, will be executed in an only machine cycle and
5 since there are 7 pipeline stages, it will cipher (resp. decipher) 7 64 bits blocks for each execution of the algorithm.

The HIPCrypto was designed to offer four kinds of configurations, ie, 1, 2, 4 or 8 integrates in
10 series (table 3).

Each pipeline segment is executed in one clock cycle. For one chip configuration, seven 64 bits blocks are processed each 56 (7 x 8) machine cycles. For 2 chips seven 64 bits blocks are processed each 28 (7 x 4)
15 machine. For 4 chips configuration seven 64 bits blocks are processed each 14 (7 x 2) machine cycles. For 8 chips configuration seven 64 bits blocks are processed each 7 (7 x 1) machine cycles, that is to say, one 64 bits block for each machine cycle.

20 The proposed HIPCrypto's structure can be adapted to different uses. The adequate compromise between throughput and cost can be obtained by selecting the number of chips operating in series.

The signals used for selecting the chip
25 configurations were divided in two groups: three signals that will define the configuration cch <2:0> and three signals that will define the position of the chip into the chain pos <2:0>. Tables 4 and 5 show respectively the configurations and the possible positions.

30 The sub-keys are stored in 4 RAMs according to Figures 3 and4.

For sub-keys $Z1(i)$, $Z2(i)$, $Z3(i)$ and $Z4(i)$, a 128 bits x 8 memory is used. The first 64 bits of each memory position, least significant bits, store the cipher sub-keys (positions 0 to 63) and the last 64 bits, most significant bits (positions 64 to 127), store the decipher

5   sub-keys. The selection to execute the algorithm in cipher or decipher mode is made through the bus selection (see Figures 3 and 4).

For sub-keys $Z5(i)$ and $Z6(i)$, two 32 bits x 8

10   RAMs are used, where the 16 least significant bits (0 to 15), store the cipher sub-keys $Z5(i)$ and $Z6(i)$, and the 16 bits most significant store the decipher sub-keys.

For the sub-keys $Z1(9)$, $Z2(9)$, $Z3(9)$ and $Z4(9)$, a 64 bits x 2 memory is used.

15   Control Unit

The control unit (see Figure 3) is the operational block that controls the operation of the architecture. This unit together with some extra circuits is responsible for the generation of the control signals.

20   The main functions of this unit are described in the following.

The control unit selects ciphering and deciphering modes, i.e. sleeting the cipher and decipher sub-keys respectively in each embedded memory.

25   The control unit also allows the correct initialization, feeding and synchronization of the pipeline stages by generating all enables and reset signals for each internal block.

The output stage will only be used by the

30   last chip in each configuration. This selection is also

10

performed by the control unit through the selected
configuration for each chip.

HIPCrypto performance

       Table 7 shows some examples of the
5   performance of HIPCrypto implemented in a two metal layer
0,7 micron CMOS technology.

CLAIMS

1. A METHOD FOR THE HARDWARE IMPLEMENTATION OF THE IDEA CRYPTOGRAPHIC ALGORITHM - HIPCrypto, patented in the USA under the no. US05214703, that makes use of a
5   seven stages pipeline to be implemented as a synchronous circuit, that will be referred as micro-pipeline, coupled to an output stage as described in figure 2; so that each stage of the pipeline supplies partial results for the following stage and receives partial results from the
10 previous stage at each clock pulse of the synchronous circuit; ao that there exists a feedback from stage number 7 to stage number 1, controlled by the digital control unit so that, for each of 16 rounds for ciphering a 64 bits block, the first stage of the pipeline is fed with partial
15 results from the output of the seventh pipeline stage and that the pipeline is fed with a new block when the ciphering process is completed; and the sub-keys used in the data ciphering and deciphering processes, in agreement with the definition of IDEA the algorithm, is stored in
20 four dedicated memory units.

2. A METHOD FOR THE HARDWARE IMPLEMENTATION OF THE IDEA CRYPTOGRAPHIC ALGORITHM - HIPCrypto, in agreement with claim 1 in which the operations 1, 2, 3 and 4 of the description of the IDEA cryptographic algorithmis
25 executed by two 16 bits multiplier units and two 16 bits adder units; and these units compose pipeline stage number as described in figure 2; so that this stage receives, either a new 64 bits block from data input or a partial result from the seventh stage and the ciphering or
30 deciphering sub-keys corresponding to this stage as described in the figures 2 and 3; so that the inputs and

outputs of this stage are connected to input and output registers respectively.

3. A METHOD FOR THE HARDWARE IMPLEMENTATION
5 OF THE IDEA CRYPTOGRAPHIC ALGORITHM - HIPCrypto, in agreement with claim 1 in which the operations 5 and 6 of the description of the IDEA cryptographic algorithm are executed by two 16 bits exclusive-or units; and these units compose stage number two of the pipeline as described in
10 the figure 2; so that this stage receives partial results from the first stage as described in the figures 2 and 3; so that the inputs and outputs of this stage are coneected to input and output registers respectively.

4. A METHOD FOR THE HARDWARE IMPLEMENTATION
15 OF THE IDEA CRYPTOGRAPHIC ALGORITHM - HIPCrypto, in agreement with claim1 in which the operation 7 of the description of the IDEA cryptographic algorithm is executed by a 16 bits multiplier unit; and this unit composes stage number three of the pipeline as described in the figure 2;
20 so that this stage receives partial results from the previous stage and ciphering and deciphering sub-keys corresponding to this stage as described in figures 2 and 3; and that the inputs and outputs of this stage are connected to input and output registers respectively.

25 5. A METHOD FOR THE HARDWARE IMPLEMENTATION OF THE IDEA CRYPTOGRAPHIC ALGORITHM - HIPCrypto, in agreement with claim1 in which the operation 8 of the description of the IDEA cryptographic algorithm is executed by a 16 bits adder unit; and this unit composes stage
30 number four of the pipeline as described in the figure 2; so that this stage receives partial results of the stages 2

and 3 as described in the figure 2; and that the inputs and outputs of this stage are connected to input and output registers respectively.

5   6. A METHOD FOR THE HARDWARE IMPLEMENTATION OF THE IDEA CRYPTOGRAPHIC ALGORITHM - HIPCrypto, in agreement with claim1 in which the operation 9 of the description of the IDEA cryptographic algorithmis executed by a 16 bits multiplier unit; and this unit composes stage number five of the pipeline as described in figure 2; so

10   that this stage receives partial results from the previous stage and ciphering and deciphering sub-keys corresponding to this stage as described in the figures 2 and 3; and that the inputs and outputs of this stage are connected to input and output registers respectively.

15   7. A METHOD FOR THE HARDWARE IMPLEMENTATION OF THE IDEA CRYPTOGRAPHIC ALGORITHM - HIPCrypto, in agreement with claim1 in which the operation 10 of the description of the IDEA cryptographic algorithm is executed by a 16 bits adder unit; and this unit composes stage

20   number six of the pipeline as described in the figure 2; so that this stage receives partial results of stages 3 and 5 as described in figure 2; and that the inputs and outputs of this stage are connected to input and output registers respectively.

25   8. A METHOD FOR THE HARDWARE IMPLEMENTATION OF THE IDEA CRYPTOGRAPHIC ALGORITHM - HIPCrypto, in agreement with claim1 in which the operations 11, 12, 13 and 14 of the description of the IDEA cryptographic algorithm are executed by four 16 bits exclusive-or units;

30   and these units compose stage number seven of the pipeline as described in figure 2; so that this stage receives

partial results from stages 1, 5 and 6 as described in the figure 2; and that the inputs and outputs of this stage are connected to input and output registers respectively.

9. A METHOD FOR THE HARDWARE IMPLEMENTATION OF THE IDEA CRYPTOGRAPHIC ALGORITHM – HIPCrypto, in agreement with claim1 in which the operations 15, 16, 17 and 18 of the description of the IDEA cryptographic algorithm are executed by two 16 bits multiplier units and for two 16 bits adder units; and these units compose the output stage of the pipeline as described in figure 2; so that this stage receives partial results from stage 7 of the pipeline and ciphering and deciphering sub-keys corresponding to the output stage as described in figures 2 and 3; and that the inputs and outputs of this stage are connected to input and output registers respectively.

10. A METHOD FOR THE HARDWARE IMPLEMENTATION OF THE IDEA CRYPTOGRAPHIC ALGORITHM – HIPCrypto, in agreement with claim1, in which the sub-keys used in ciphering and deciphering process are stored in four dedicated memories according to figure 4 as follows: ciphering sub-keys $Z1(i)$, $Z2(i)$, $Z3(i)$ and $Z4(i)$ (i from 1 to 8) and deciphering sub-keys $Z1(i)$, $Z2(i)$, $Z3(i)$ and $Z4(i)$ (i from 1 to 8) stored in a 128 bits x 8 memory; ciphering sub-keys $Z5(i)$ (i from 1 to 8) and deciphering sub-keys $Z5(i)$ (i from 1 to 8) stored in the first 32 bits x 8 memory; ciphering sub-keys $Z6(i)$ (i from 1 to 8) and deciphering sub-keys $Z6(i)$ (i from 1 to 8) stored in the second 32 bits x 8 memory; ciphering sub-keys $Z1(9)$, $Z2(9)$, $Z3(9)$ and $Z4(9)$ and deciphering sub-keys $Z1(9)$, $Z2(9)$, $Z3(9)$ and $Z4(9)$ stored in the 64 bits x 2 memory.

11. A METHOD FOR THE HARDWARE IMPLEMENTATION OF THE IDEA CRYPTOGRAPHIC ALGORITHM - HIPCrypto, in agreement with claims 1 and 2, in which a second pipeline level, denominated macro-pipeline, allows the concatenation of 2, 4 or 8 circuits operating with a micro-pipeline of seven stages as indicated in the table 3.

12. A METHOD FOR THE HARDWARE IMPLEMENTATION OF THE IDEA CRYPTOGRAPHIC ALGORITHM - HIPCrypto, in agreement with claims 1 and 2, in which " first-in first-out "(FIFO) memories are used in order to synchronize the data provenient from non adjacent stages in the following way: a 64 bits x 5 positions FIFO connecting stages 1 and 7, a 16 bits x 2 positions FIFO connecting stages 3 and 6, a 16 bits x 1 position FIFO connecting stages 2 and 4, a 16 bits x 1 position FIFO connecting stages 5 and 7, as described in the figure 3.

DRAWINGS

Table 1

DES Algorithm velocity implemented in different plataforms.

Plataforms	Frequency (MHz)	Troughput (cipher)
8088	4,7	23,68 Kbps
68000	7,6	57,6Kbps
80286	6	70,4 Kbps
68020	16	224 Kbps
68030	16	249,6 Kbps
80386	25	320 Kbps
68030	50	640 Kbps
68040	25	1,024 Mbps
68040	40	1,472 Mbps
80486	66	2,752 Mbps
HP900/887	125	10,816 Mbps
Sun ELC		1,664 Mbps
HyperSparc		2,048 Mbps
RS6000-350		3,392 Mbps
Sparc 10/52		5,376 Mbps
DEC Alpha 4000/610		9,856 Mbps

Table 2

Commercial integrates of DES Algorithm

Fabricant	Integrate	Fabrication year	Frequency	Troughput
AMD	Am9518	1981	3 MHz	10,4 Mbps
AMD	Am9568	?	4 MHz	12 Mbps
AMD	AmZ8068	1982	4 MHz	13,6 Mbps
AT&T	T7000A	1985	?	15.2 Mbps
CE-infosys	SuperCrypt CE99C003	1992	20 MHz	100 Mbps
CE-infosys	SuperCrypt CE99C003A	1994	30 MHz	160 Mbps
Cryptech	Cry12C102	1989	20 MHz	22,4 Mbps
Newbridge	CA20C03A	1991	25 MHz	30,8 Mbps
Newbridge	CA20C03W	1992	8 MHz	5,12 Mbps
Newbridge	CA95C68/18/09	1993	33 MHz	117,4 Mbps
Pijnenburg	PCC100	?	?	20 Mbps
Semaphore Communications	Roadrunner284	?	40 Mhz	284 Mbps
VLSI technology	VM007	1993	32 MHz	160 Mbps
VLSI technology	VM009	1993	33 MHz	112 Mbps
VLSI technology	6868	1995	32 MHz	512 Mbps

Table 3

Configuration	Iteration numbers for configuration
1 integrate	8 iterations
2 integrates	4 iterations in each integrate
4 integrates	2 iterations in each integrate
8 integrates	1 iteration in each integrate

Table 4:

ch2	ch1	ch0	Configuration
			1 integrate.
			2 series-connected integrates.
			4 series-connected integrates.
			8 series-connected integrates.

Table 5:

pos2	pos1	pos0	Integrate position.
0	0	0	First position
0	0	1	Second position
0	1	0	Third position
0	1	1	Fourth position
1	0	0	Fifth position
1	0	1	Sixth position
1	1	0	Seventh position
1	1	1	Eighth position

Table 6:

Configuration			Position		
cch2	cch1	cch0	pos2	pos1	pos0
0	0	0	0	0	0
0	0	1	0	0	0
			0	0	1
0	1	0	0	0	0
			0	0	1
			0	1	0
			0	1	1
1	0	0	0	0	0
			0	0	1
			0	1	0
			0	1	1
			1	0	0
			1	0	1
			1	1	0
			1	1	1

Table 7:

Macro-pipeline Configurations	Performance Clock frequency: (59 MHz)
1 integrate	424 Mbps
2 integrates	848 Mbps
4 integrates	1,7 Gbps
8 integrates	3,4 Gbps
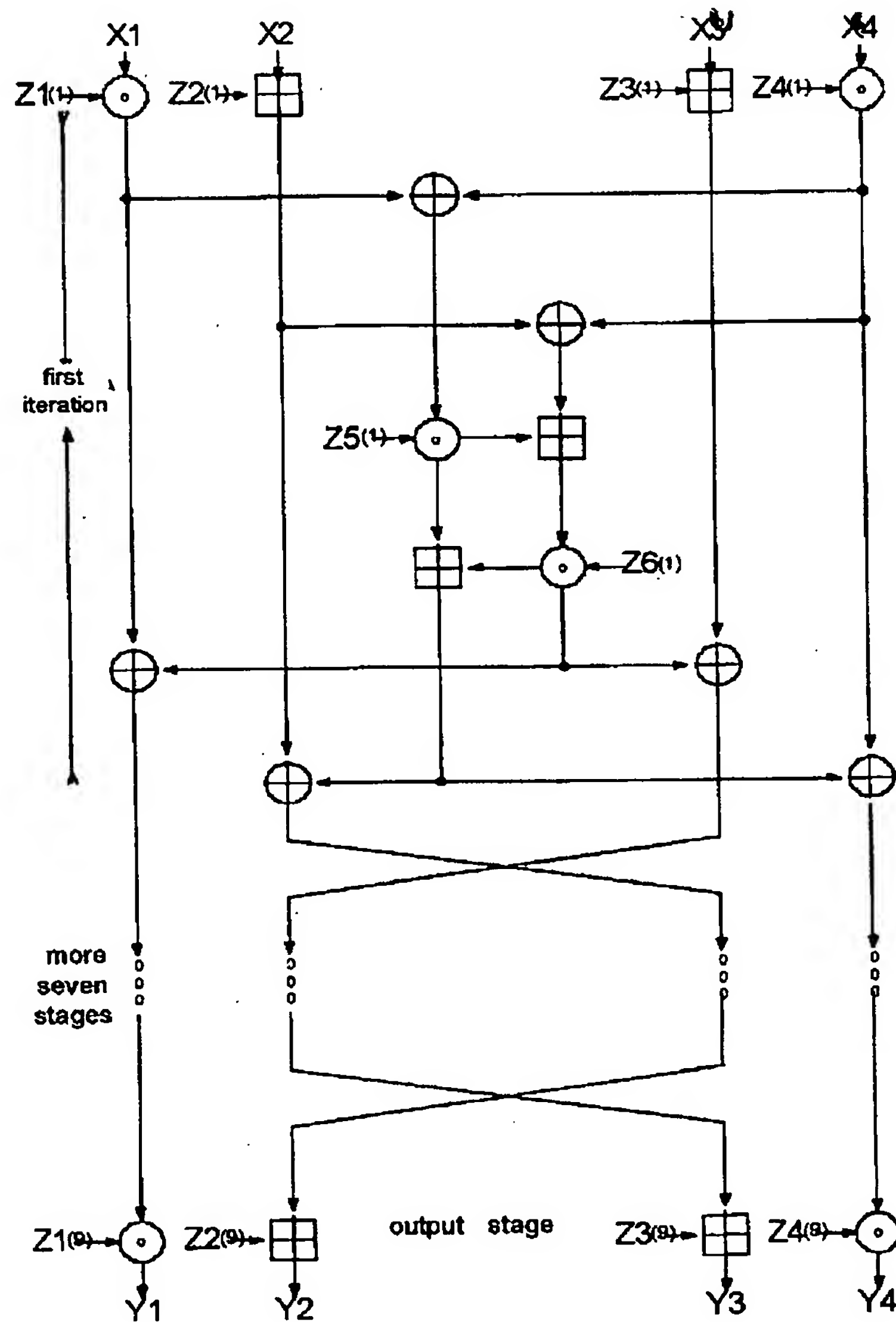
Figure 1

Figura 2

7/8

Figure 3

Architecture

DATA OUTPUT
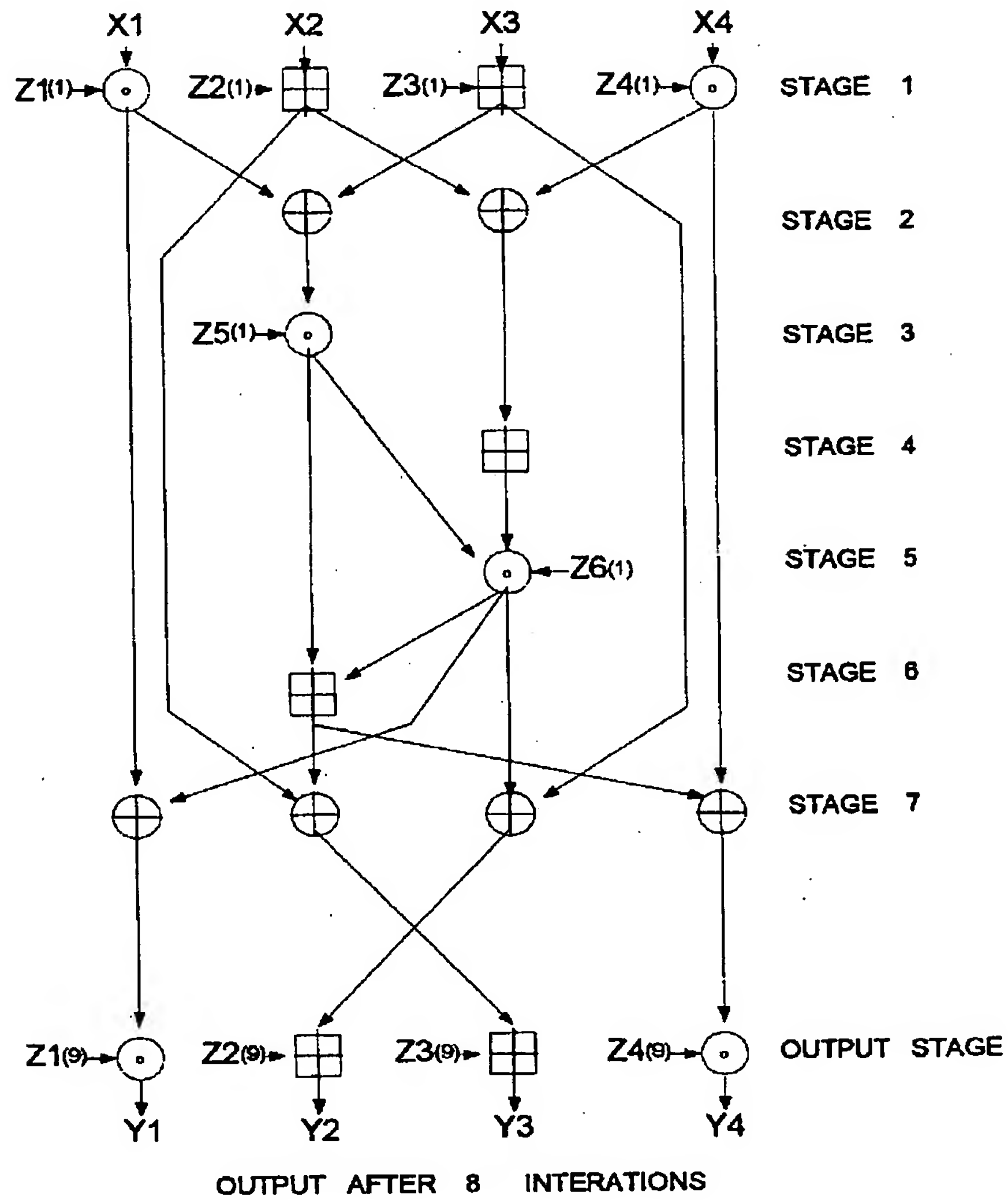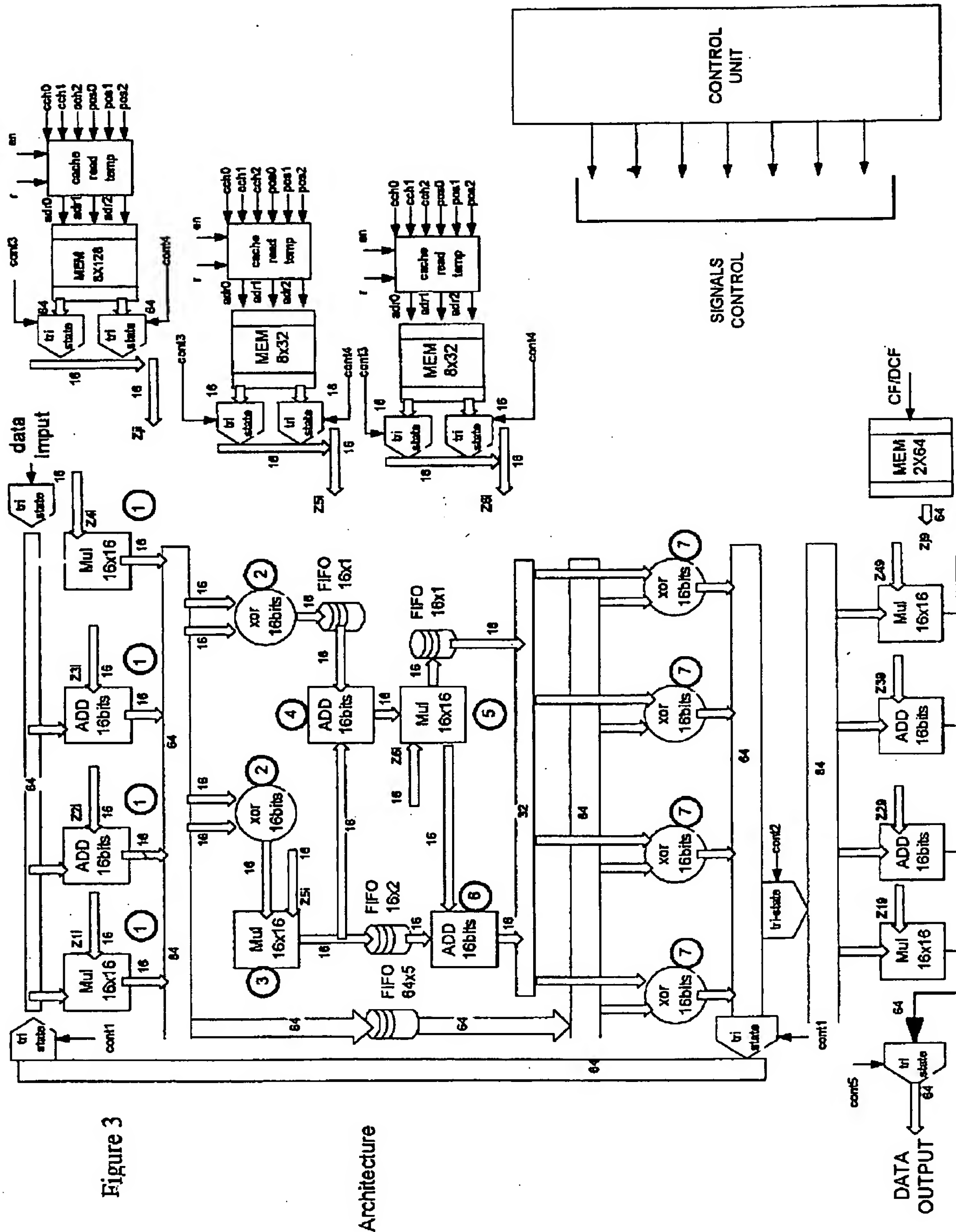
Figure 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/BR 99/00076

CLASSIFICATION OF SUBJECT MATTER

IPC[7]: H 04 K 1/04

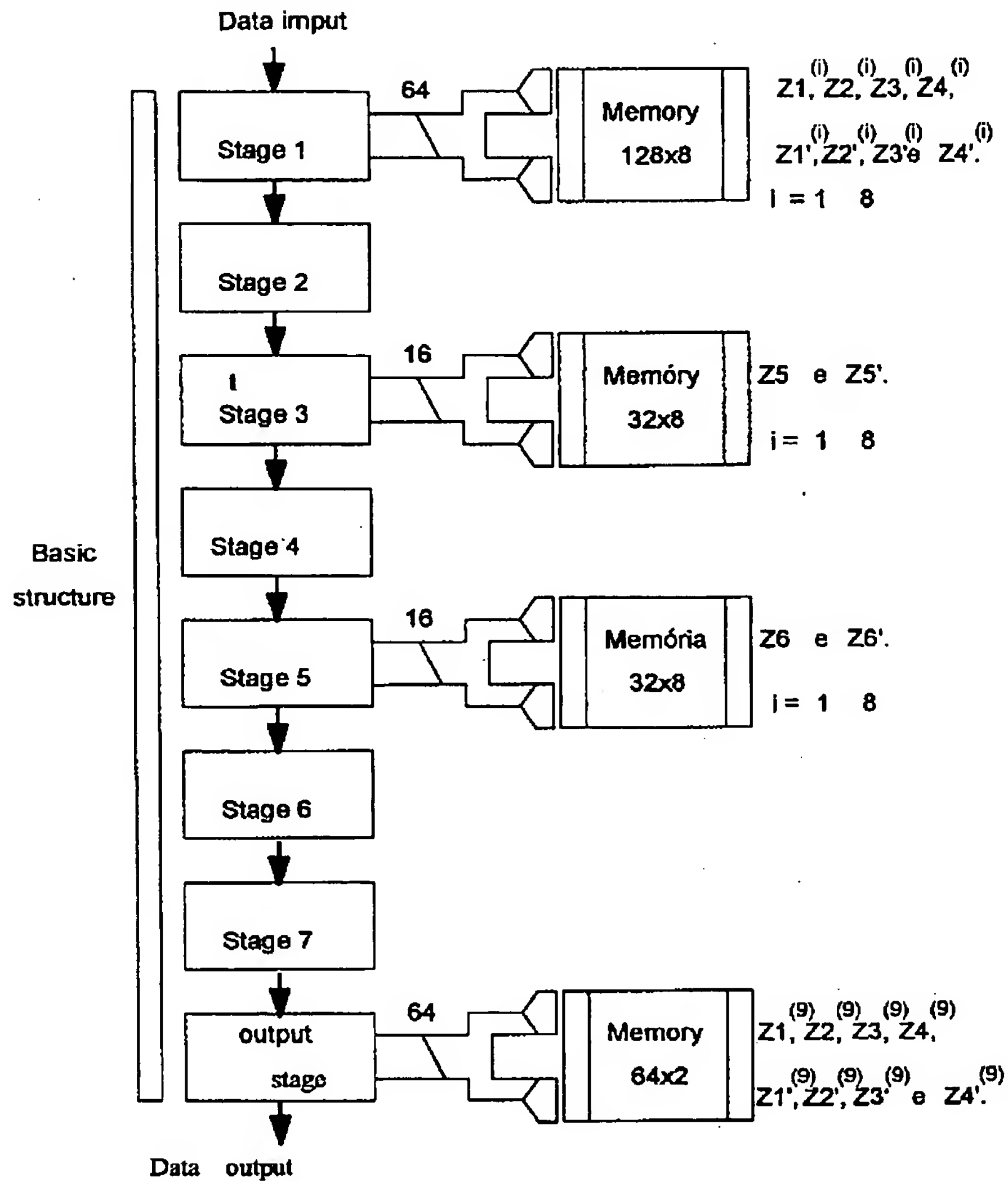According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC[7]: H 04 K 1/00, 1/04

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WPI

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5214703 A (MASSEY) 25 May 1993 (25.05.93) totality.	1-12
A	US 5161193 A (LAMPSON) 3 November 1992 (03.11.92) abstract; claims; fig.1,2,3.	1-12
A	US 5020106 A (RABOLD) 28 May 1991 (28.05.91) abstract; claims; column 1, lines 5-52; fig.1,2.	1-12
A	US 4850019 A (SHIMIZU) 18 July 1989 (18.07.89) abstract; claims; fig.1-11.	1-12
A	US 4668103 A (WILSON) 26 May 1987 (26.05.87) abstract; claims; fig.1-4.	1-12
A	US 4275265 A (DAVIDA) 23 June 1981 (23.06.81) abstract; claims; fig.1,2.	1-12

☐ Further documents are listed in the continuation of Box C.   ☒ See patent family annex.

* Special categories of cited documents:

„A" document defining the general state of the art which is not considered to be of particular relevance

„E" earlier application or patent but published on or after the international filing date

„L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

„O" document referring to an oral disclosure, use, exhibition or other means

„P" document published prior to the international filing date but later than the priority date claimed

„T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

„X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

„Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

„&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
8 May 2000 (08.05.2000)	31 July 2000 (31.07.2000)

Name and mailing adress of the ISA/AT	Authorized officer
Austrian Patent Office Kohlmarkt 8-10; A-1014 Vienna	Erber
Facsimile No. 1/53424/535	Telephone No. 1/53424/382

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/BR 99/00076

Patent document cited in search report			Publication date	Patent family member(s)			Publication date
US	A	5214703	25-05-1993	AT	E	91208	15-07-1993
				DE	C0	59100171	05-08-1993
				EP	A1	482154	29-04-1992
				EP	B1	482154	30-06-1993
				ES	T3	2042346	01-12-1993
				HK	A1	1003338	23-10-1998
				JP	T2	5500121	14-01-1993
				WO	A2	9118459	28-11-1991
				WO	A3	9118459	05-03-1992
US	A	5161193	03-11-1992	CA	AA	2044860	30-12-1991
				EP	A2	464565	08-01-1992
				EP	A3	464565	26-05-1993
				JP	A2	6077954	18-03-1994
				JP	B2	2693881	24-12-1997
				US	A	5099517	24-03-1992
				US	A	5235644	10-08-1993
US	A	5020106	28-05-1991	DE	C0	59007408	17-11-1994
				EP	A1	382680	16-08-1990
				EP	B1	382680	12-10-1994
US	A	4850019	18-07-1989	DE	C0	3689510	17-02-1994
				DE	T2	3689510	19-05-1994
				DE	C0	3650365	14-09-1995
				DE	T2	3650365	08-02-1996
				EP	A2	221538	13-05-1987
				EP	A3	221538	10-08-1988
				EP	A2	454187	30-10-1991
				EP	A3	454187	29-04-1992
				EP	B1	221538	05-01-1994
				EP	B1	454187	09-08-1995
				JP	A2	62109083	20-05-1987
				JP	B4	7060292	28-06-1995
				JP	A2	62113191	25-05-1987
				JP	B4	3033269	16-05-1991
US	A	4668103	26-05-1987	US	A	4520232	28-05-1985
US	A	4275265	23-06-1981			none	

PCT/ISA/210 (patent family annex) (July 1998)